Coordinated Sampling: An Efficient, Network-Wide Approach for Flow Monitoring

Vyas Sekar, Michael K. Reiter,
Walter Willinger [1], Hui Zhang

Jul 16, 2007
CMU-CS-07-139

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

[1] AT&T Labs–Research

This research was supported in part by National Science Foundation grant number CNS-0433540 and ANI-0331653 and U.S. Army Research Office contract number DAAD19-02-1-0389. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of NSF, ARO, Carnegie Mellon University, or the U.S. Government or any of its agencies.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
16 JUL 2007		**00-00-2007 to 00-00-2007**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Coordinated Sampling: An Efficient, Network-Wide Approach for Flow Monitoring	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Carnegie Mellon University,School of Computer Science,Pittsburgh,PA,15213**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

We present Coordinated Sampling, a new technique for improved flow-level monitoring. Our approach derives from three key design decisions: flow sampling instead of uniform packet sampling; hash-based flow selection to achieve coordination between routers without needing explicit communication channels; and an approach for distributing responsibilities across routers to achieve network-wide monitoring objectives while taking into account resource constraints on each router. We demonstrate that Coordinated Sampling presents an attractive solution for ISPs. First, it more than doubles flow coverage to support security applications and does so without compromising the accuracy of traditional traffic engineering applications. Second, it enables network operators to directly specify and achieve fine-grained network-wide monitoring objectives. Third, it naturally load balances monitoring responsibilities across routers and at the same time efficiently leverages the available capacity on each router.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**29**	

Keywords: Network Monitoring, Sampling, Flow collection

Abstract

We present Coordinated Sampling, a new technique for improved flow-level monitoring. Our approach derives from three key design decisions: flow sampling instead of uniform packet sampling; hash-based flow selection to achieve coordination between routers without needing explicit communication channels; and an approach for distributing responsibilities across routers to achieve network-wide monitoring objectives while taking into account resource constraints on each router. We demonstrate that Coordinated Sampling presents an attractive solution for ISPs. First, it more than doubles flow coverage to support security applications and does so without compromising the accuracy of traditional traffic engineering applications. Second, it enables network operators to directly specify and achieve fine-grained network-wide monitoring objectives. Third, it naturally load balances monitoring responsibilities across routers and at the same time efficiently leverages the available capacity on each router.

1 Introduction

Many network management and traffic engineering applications depend on flow-level [2] data collected by routers. While prior work has demonstrated the benefits of using such measurements for traffic engineering and customer accounting applications (e.g., [15, 14, 10]), there is still a fundamental disconnect between the goals of network management applications and the monitoring primitives implemented in routers.

• First, network operators would like to specify and achieve *network-wide* monitoring objectives. However, existing solutions, including recent work on data streaming algorithms (e.g., [23, 26]), are designed as single router solutions. Since these solutions operate from the perspective of a single vantage point, they do not provide a way for network operators to directly specify and achieve network-wide measurement goals.

• Second, flow-level measurements are being increasingly used in many security applications including network anomaly detection (e.g., [24]), identification of unwanted application traffic (e.g., [8]), and the detection and forensic analysis of worm and DDoS attacks (e.g., [43, 38]). This changing scope of the applications that use flow data has given rise to concerns (e.g., [32, 5]) regarding the fidelity of traditional packet sampling based techniques. Specifically, these applications benefit from greater *flow coverage*. In contrast to traditional traffic engineering and accounting applications, which only need an aggregate traffic volume estimate, in these new applications it is necessary to identify and analyze as many distinct flows that make up the total traffic as possible. Current sampling techniques lack this ability.

• Third, the available monitoring capacity on each router is bound by technological resource constraints. Network operators would like to optimally leverage as much of the available monitoring capacity on routers as possible. However, in existing solutions routers operate in isolation, with each device independently recording a subset of the traffic it observes. Such an approach is not only inefficient in terms of utilizing the router resources, but also raises concerns for network operators in having to deal with redundant and possibly ambiguous measurements from multiple routers[1].

We present *Coordinated Sampling*: a technique for efficient, network-wide flow-level monitoring. Coordinated Sampling allows operators to specify and achieve network-wide monitoring goals while optimally leveraging the measurement capabilities of each router. Our approach derives from the following design primitives: flow sampling [20], hash-based selection [47], and network-wide optimization [7].

• By using flow sampling [20] instead of packet sampling, Coordinated Sampling provides better flow coverage by avoiding the bias of packet sampling against small flows. At the same time, using flow sampling does not affect the fidelity of traffic volume estimation, and thus does not compromise the accuracy of traditional traffic engineering applications.

• Since both router memory and reporting bandwidth are scarce resources, coordinating measurements along a single routing path can better utilize the available monitoring capacity in the network by eliminating duplicated measurement effort. Coordinated Sampling uses hash-based selection to coordinate measurements across routers without requiring explicit communication between routers.

[1] Some ISPs prefer to have Netflow-like capabilities enabled only in a small subset of routers for this reason [15].

• Coordinated Sampling provides an optimization framework to specify and achieve network-wide monitoring objectives under real-world resource constraints on routers. An optimal solution can be directly translated into a *sampling manifest* for individual routers in the network. The sampling manifest specifies the set of traffic flows that a router is required to record and report.

We evaluate the benefits of Coordinated Sampling over a wide range of network topologies. We get a two-fold increase in flow coverage compared with uniform packet sampling. On a more fine-grained flow coverage metric, the minimum fractional coverage per OD-flow (Section 3.3.1), Coordinated Sampling provides an order of magnitude improvement over other sampling alternatives. We explore the robustness aspects of Coordinated Sampling and show that our scheme is robust with respect to errors in input data and realistic changes and uncertainties in traffic demands.

ISPs can derive several additional operational benefits from Coordinated Sampling. Coordinated Sampling naturally load balances monitoring functionality across the network, thereby avoiding the occurrence of reporting hotspots. By minimizing duplicated measurements, Coordinated Sampling reduces the management overhead of merging data collected from multiple monitors [12]. We also show that our approach is general and flexible enough to facilitate a wide variety of network management applications. The combination of design principles underlying Coordinated Sampling has far-reaching implications for enabling more centralized management and operations of ISPs [3, 6, 19].

The rest of the paper is organized as follows. We review related work in the next section. In Sections 3 and 4 we present a detailed description of our approach (including the formulation of the network-wide monitoring optimization problem and the implementation of such a coordinated monitoring approach), and discuss some of the practical issues associated with deploying our scheme. In Section 5 we demonstrate the benefits of Coordinated Sampling over currently used sampling techniques and also show that our scheme is robust under real-world networking conditions. We summarize our main results and discuss interesting avenues of future work in Section 6.

2 Related Work

Prior work has stressed the need for taking a more network-wide approach for traffic engineering [15, 45] and network diagnosis [24, 25, 30].

Cantieni et al. [7] consider the problem of optimally configuring uniform packet sampling rates in a network. The constrained optimization formulation in their work shares some structural similarity to our approach in Section 3.3.1. However, there are two key differences in our approaches. First, our focus on flow coverage is motivated by the security applications we envision, but does not compromise or impair the accuracy of the resulting data for the more traditional traffic engineering applications they consider. Second, while it is reasonable to assume that the probability of a single packet being sampled multiple times across routers is negligible, the assumption is not valid for flow-level monitoring. We rely on coordination as a design primitive to avoid duplicate flow reporting to maximize flow coverage and to specify more fine-grained coverage objectives.

While coordination to minimize redundancy is a common high-level theme between Coordinated Sampling and the approach of Sharma and Byers [39], our work differs in a number of

significant ways. First, our network-wide approach is more general than the specific goal of minimizing redundant monitoring. Second, by relying on hash-based sampling we achieve coordination *without* explicit communication, while their approach potentially requires every pair of routers in the network to periodically exchange snapshots of the set of flows they are currently monitoring. Third, our formulation for obtaining the optimal sampling strategy takes into account resource constraints on routers and can be generalized to handle heterogeneity across routers.

A well-known application of hash-based packet selection [47] is trajectory sampling [9, 27]. In the case of trajectory sampling, the measurement objective is to ensure that all routers observe a specific subset of packets. Thus, all routers are assigned the same hash range to reveal packet trajectories through the network. In contrast, Coordinated Sampling uses hash-based sampling for exactly the opposite functionality: to ensure that different routers monitor different flows.

Other related efforts in this problem space concern improvements or redesigns of single-router sampling algorithms: adapting the packet sampling rate to changing traffic conditions for tuning the processing, memory, and reporting bandwidth overheads (e.g., [13, 22]); tracking flows with high traffic counts (elephant flows) with high accuracy [14]; obtaining better traffic estimates from sampled measurements [20, 10]; reducing the overall amount of measurement traffic [11]; and data streaming algorithms for specific applications (e.g., [23, 26]). These approaches focus on single-router solutions and lack the network-wide view that Coordinated Sampling provides. Additionally, these solutions either lack generality across applications (e.g., different traffic metrics require specialized streaming algorithms) or may in fact be counter-productive in the context of flow coverage. For example, techniques for tracking flows with high traffic counts [14, 11] are attractive single-router solutions for traffic engineering and customer accounting. However, keeping track of elephant flows will increase redundant monitoring across routers (every router tracks the same set of elephant flows), without increasing flow coverage.

3 Coordinated Sampling: Design

In this section, we present the three design primitives underlying Coordinated Sampling. Our discussion assumes the common 5-tuple (*srcIP, dstIP, srcport, dstport, protocol*) notion of a IP flow.

3.1 Flow sampling

Due to heavy-tailed flow-size distributions [23] observed in real traffic, the flow coverage provided by uniform packet sampling is poor. Selecting random flows, rather than packets, can improve flow coverage by avoiding sampling biases due to heavy-tailed distributions. For completeness, we briefly describe the conceptual implementation of *flow sampling* [20]. As each packet arrives, the router computes a flow label on the packet header. This flow label can be a hash function computed on the 5-tuple used for identifying the flow. Each router maintains a table of the flows it is currently monitoring in its $Flowtable$. If the flow already exists in the table, the router updates the byte and packet counters corresponding to the entry. Otherwise, it is a previously unrecorded flow, and the router selects it with sampling probability s (e.g., if the computed hash value falls within

a range of size s). The implementation of flow sampling requires the same packet processing and table lookup capabilities as Sample and Hold [14]. However, there is one key difference between the two techniques. Since the focus of Sample and Hold is to identify and maintain near-exact counts of elephant flows, the algorithm picks random packets from the packet stream to create a new entry in the $Flowtable$. To obtain better flow coverage, flow sampling selects flows at random with probability s.

3.2 Hash-based coordination

If each router operates in isolation, i.e., independently recording a subset of flows it observes, the resulting measurements are likely to contain duplicates. This implies a potentially significant waste of reporting bandwidth and the memory resources on routers[2] and puts more stress on these already constrained resources. Further, the resulting multiplicity can cause additional data management overhead when merging or analyzing the information collected from multiple monitoring points. By adopting coordination as a design primitive, we can largely eliminate these disadvantages. One approach for coordination would be to enable explicit communication among the routers on the same router-level path, either in the form of specialized inter-router message exchanges (e.g., [4, 33, 39]), or through packet marking schemes (e.g., [37, 28]).

We propose an alternative approach that relies on *hash-based* selection for implementing coordination among routers without requiring explicit communication. Specifically, we use hash-based selection so that different routers on the same router-level path (between a network ingress and egress) select distinct flows. Typically, the hash function is computed on the invariant fields in packet headers [9, 41][3]. The key is to assign non-overlapping ranges of the hash-space to the different routers on the path. Each router computes the hash of the IP 5-tuple of the packet. The router only selects and records flows that belong to its assigned hash-range. Since the hash-ranges do not overlap, the sets of flows recorded across routers are mutually non-overlapping.

3.3 Network-wide optimization

The goal of a network monitoring system can be typically expressed as a network-wide objective; for example, maximizing the total flow coverage or providing guarantees on flow coverage for specific subsets of the total traffic. By taking a network-wide approach, we can optimally satisfy an ISP's monitoring objective, while operating within the resource constraints of individual routers, and taking into account possible heterogeneities in router capacities. We present an optimization framework that allows network operators to directly translate their network-wide monitoring objectives into per-router configurations.

[2]As observed by Estan and Varghese [14] using flow-level sampling requires access to fast SRAM, as opposed to uniform packet sampling which can work with slow DRAM.

[3]Invariant fields are those that do not change along a router-level path, e.g., the IP 5-tuple representing the flow record; in contrast, fields such as the TTL and the checksum are not invariant.

3.3.1 Assumptions and notation

Staying within the confines of an ISP, our proposed model of Coordinated Sampling assumes that a centralized network operations center (NOC) has access to the ISP's routing and traffic matrices. Based on this information, the NOC computes the optimal sampling strategy and distributes *sampling manifests* to individual routers. The sampling manifest is a configuration file that specifies the subset of traffic flows (in terms of a hash output range) that the router is supposed to record and report to the NOC. Note that such a centralized approach is consistent with the operating model of modern ISPs, where operators push out router configuration files (e.g., routing tables, ACLs) and collect information from the routers.

A natural formulation for such network management problems is in terms of *Origin-Destination (OD) flows*. Each OD-flow is characterized by a network ingress point, a network egress point, the total traffic (e.g., number of bytes, packets, or IP-level flows), and the router-level path(s) that the OD-flow takes. We make two simplifying assumptions in our formulation. First, we assume that the traffic matrix (number of IP flows per OD-flow) and routing information for the network are given and that these change infrequently. Second, we assume that each OD-flow is characterized by a single router-level path.

Let $i = 1, \ldots, M$ denote the set of OD-flows in the network. Each OD-flow i is characterized by its router-level path. The traffic on OD-flow i is given in terms of the number P_i of distinct IP-level flows (e.g., per five minute interval) that make up the OD-flow. Let $j = 1, \ldots, N$ denote the set of routers in the network. We introduce variables d_{ij} to denote the fraction of traffic (in terms of IP-level flows) of OD-flow i that is monitored by router j. Note that if router j does not lie on the path of OD-flow i, then the variable d_{ij} will not appear in the formulation.

3.3.2 Constrained optimization

The high-level goal of this optimization framework is to maximize the network-wide monitoring objective (e.g., total flow coverage), subject to the per-router resource constraints.

As in Sample and Hold [14] and other approaches that are similar to flow sampling (e.g., [23]), we do not model the packet processing constraints of routers, since we assume that by keeping the flow counters in SRAM it is feasible to implement such capabilities. The only resource constraints then are (a) memory (per-flow counters in SRAM) and (b) bandwidth for reporting the flow records to a collection point (typically the NOC). We abstract (a) and (b) into a single resource constraint R_j that represents the number of flows router j can record and report (again, per each five minute measurement interval).

If R_j denotes the sampling load constraint for router j ($j = 1, \ldots, N$), then we want to ensure that the total sampling load for router j, in terms of the total number of IP flows it is required to monitor, does not exceed the load constraint R_j. That is,

$$\forall j, \sum_i (d_{ij} \times P_i) \leq R_j \tag{1}$$

Next, for $i = 1, \ldots, M$, let $Coverage_i$ denote the fraction of traffic on OD-flow i that has been monitored. We only consider sampling manifests that ensure that routers on the path of a given

OD-flow will cover distinct IP-level flows. Thus, the fraction of traffic of OD-flow i that has been covered throughout the network is simply the sum of the fractional coverages d_{ij} of the different routers on the router-level path for OD-flow i,

$$\forall i,\ Coverage_i = \sum_j d_{ij} \tag{2}$$

Since the coverage values represent fractional quantities, we have the natural constraints:

$$\forall i,\ Coverage_i \leq 1 \tag{3}$$

Finally, since the d_{ij} define fractional coverages, they are constrained to be in the range $[0, 1]$; however, since the above constraints (Eq. 3) subsume the upper bound constraint on the d_{ij}, we are left with the non-negativity constraints on the variables d_{ij}, i.e.,

$$\forall i,\ \forall j,\ d_{ij} \geq 0 \tag{4}$$

Subject to these sets of constraints and given the input data P_i ($i = 1, \ldots, M$) and R_j ($j = 1, \ldots, N$), our objective is to maximize the benefit we obtain from the individual flow coverage values $Coverage_i$. We can define this benefit in terms of either the total coverage across OD-flows ($\sum_i P_i \times Coverage_i$) or the minimum fractional coverage per OD-flow ($\mathbf{min}_i\{Coverage_i\}$). We consider a combination of these two benefit functions and obtain a solution for our constrained optimization problem that maximizes total coverage subject to ensuring the optimal minimum fractional coverage.

We achieve this combined objective by first obtaining the solution (satisfying Eq. 1–4) that is optimal for the minimum fractional coverage objective. Denoting this optimal objective function value by $OptMinFrac$, we then introduce the additional constraints of the form

$$\forall i,\ Coverage_i \geq OptMinFrac \tag{5}$$

and proceed to obtain the solution that is optimal for the total traffic coverage objective under all of (1)–(5). Performing this two-step optimization procedure yields a solution $d^* = \langle d^*_{ij} \rangle_{1 \leq i \leq M, 1 \leq j \leq N}$ that maximizes the total flow coverage subject to achieving optimal minimum fractional coverage.[4]

3.3.3 Per-router sampling manifests

The final step of our approach consists of mapping the optimal solution into a *sampling manifest* for each router that specifies the monitoring responsibility for the router. Figure 1 presents the procedure for translating the optimal solution d^* into a sampling manifest. The sampling manifest specifies a distinct, non-overlapping, hash-range for each OD-flow traversing the router.

The resulting sampling manifests ensure that the set of IP-level flows monitored by each router on the path of the corresponding OD-flow are necessarily distinct from one another. Once a router

[4]Theoretically, this two-step approach might provide lower total flow coverage than if we optimized the total traffic coverage alone. However, in our evaluations, we find that this reduction is negligible (less than 0.1%).

GENERATESAMPLINGMANIFEST($d^* = \langle d^*_{ij} \rangle$)

1 **for** $i = 1, \ldots, M$ **do**
2 $\quad Range \leftarrow 0$
3 $\quad$ **for** $j = 1, \ldots, N$ **do**
4 $\quad\quad HashRange(i,j) \leftarrow [Range, Range + d^*_{ij}]$
5 $\quad\quad Range \leftarrow Range + d^*_{ij}$
6 $\forall j, Manifest(j) \leftarrow \{\langle i, HashRange(i,j)\rangle | d^*_{ij} > 0\}$

Figure 1: **Translating the optimal solution into a sampling manifest for each router**

COORDINATEDSAMPLING($pkt, Manifest$)

// j is the router identifier
// $Manifest = \langle i, HashRange(i,j)\rangle$
1 $OD \leftarrow$ GETODFLOWID(pkt)
// HASH returns a value in $[0,1]$
2 $h_{pkt} \leftarrow$ HASH(FLOWHEADER(pkt))
3 **if** $h_{pkt} \in Hashrange(OD, j)$ **then**
4 $\quad$ Create an entry in $Flowtable$ if none exists
5 $\quad$ Update byte and packet counters for the entry

Figure 2: **Coordinated Sampling on each router**

has received its sampling manifest, the algorithm for Coordinated Sampling that each router implements is simple (Figure 2). For each packet it observes, the router first identifies the OD-flow. Then, it computes a hash on the flow headers (e.g., the IP 5-tuple) and checks if the hash value lies in the assigned hash range for the specific OD-flow the packet belongs to (the function HASH returns a value in the range $[0,1]$). Each router maintains a $Flowtable$ of the set of flows it is currently monitoring. If the packet has been selected, then the router either creates a new entry (if none exists) or simply updates the counters for the corresponding entry in the $Flowtable$.

3.4 Generality of our approach

The optimization formulation offers much flexibility in terms of modeling router constraints, incorporating traffic and routing policies, and specifying objectives. It is easy to account for heterogeneity in routers in the network (in terms of capacity, memory, reporting bandwidth). Not only is it possible to take into account that different versions of router software and hardware may have different logging capabilities, but operators can also use the proposed formulation to specify separate sampling regimes for different classes of routers in the network (e.g., access vs. edge vs. backbone). Since our approach makes no a priori assumptions regarding the nature of the input data (i.e., internal routing and OD-traffic demands), it is general enough to accommodate arbitrary routing policies and OD-traffic matrices. For example, adding multi-path routing for each OD-flow

simply requires information about what fraction of a given OD-flow traverses each path. Also, our framework can accommodate a wide range of benefit functions (e.g., weighted combinations of the $Coverage_i$ values). However, in this paper, we restrict ourselves to the two-step combination of the total and minimum fractional coverage discussed earlier.

4 Implementation Issues

In this section, we discuss a number of practical issues related to the implementation of Coordinated Sampling.

4.1 OD-flow identification

We require that each router, on observing a packet, can identify the OD-flow to which the packet belongs (Figure 2). To enable OD-flow identification, we envision that each packet carries as part of its header its OD-flow identifier. In practice, an ISP's border routers can mark each incoming packet with this OD-flow information by determining the ingress and egress PoPs of the packet. One concern is that techniques requiring modifying packet headers or adding information to the IP-packet header (e.g., traceback [37, 28], capabilities [44]) have not been easy to deploy. The key difference is that our approach is specifically designed for deployment within a single ISP. The required OD-flow identifier modification to the packet header has only local significance within the ISP and such information can be obtained with low computational overhead [15]. While this a valid concern, we believe that the deployment barrier for Coordinated Sampling will be substantially lower compared to such schemes that require routers to perform moderately expensive computations to determine packet markings and that the markings retain their semantics across ISP boundaries [37, 28, 44].

4.2 Routing and Traffic Matrices

Prior work suggests that it is reasonable to assume that such routing information is available to network operators [15]. Further, if we do observe a shift toward more centralized network management solutions [3, 6, 19], the problem of obtaining up-to-date routing information becomes easier.

Similarly, there already exist known efficient methods for estimating traffic matrices [45, 46]. However, since these traffic matrices are estimated from possibly incomplete measurements they are likely to have estimation errors. We address this issue in Section 4.5. Traffic matrices are also known to change over time and we address this issue in Section 4.6.

4.3 Computing the optimal solution

The optimization is a linear programming formulation; obtaining an optimal solution is computationally tractable. For our evaluations, we relied on a commercial LP-solver (`CPLEX`) to compute optimal solutions. The time taken to generate the optimal solution is small: only a few seconds for

the PoP-level topologies we use in our evaluation. For example, with the largest PoP-level topology in our evaluation with 115 nodes (AS 7018) it takes 3.9 seconds on a server-range machine (Intel Xeon 2.80 GHz 4-CPU, 4 GB RAM) to generate the optimal solution. For larger router-level topologies (both synthetic and inferred [42] topologies), of the order of 200-400 nodes, computing the optimal solution takes between 30-90 seconds.

4.4 Per-router processing

One concern is the per-packet processing required on each router (e.g., computing the hash-function, performing $Flowtable$ lookups, and updating counters). However, prior work [14, 41, 40] has demonstrated that it is indeed feasible to implement such per-packet processing capabilities in router hardware without much overhead. Also, modern routers already perform many per-packet operations for forwarding, and such processing functionality is typically implemented using highly parallelized hardware circuitry.

Flow vs. Packet sampling: The requirements for flow sampling as opposed to packet sampling are well understood in the literature [14]; packet sampling only needs to process a subset of packets whereas flow sampling needs to process every packet. Netflow-style packet sampling is constrained by packet processing capabilities since it uses (slow) DRAM for updating counters. However, by using counters in (faster) SRAM flow sampling becomes feasible even at high line rates [14]. In our evaluation, we assume that each PoP-level node can track 200,000 flow counters. Even assuming a conservative estimate of 32 bytes for each flow entry [14], this translates into a requirement of only $200,000 \times 32 = 6.4$ MB of SRAM per PoP, which is well within the reach of modern router hardware.

Hash-functions: We use hash-based flow sampling to achieve coordination without explicit communication. There are ongoing efforts between router vendors and IETF working groups [47] to standardize hash-function implementations and support hash-based sampling as a basic primitive in routers. Since the requirements on the type of hash-functions we desire are quite simple [41, 9] (e.g., we need no strong cryptographic guarantees), they are amenable to fast hardware implementations [34]. In our current implementation we use the BOB hash function recommended by Zseby et al. [47].

4.5 Robustness to input errors

Available OD-level traffic matrices are typically obtained using estimation techniques (e.g., [45, 46]) and as such represent only an approximation of the actual OD-traffic demands. Keeping the rest of the assumptions the same, we are interested in the sensitivity of our approach to inaccuracies of estimated OD-traffic matrices.

To be more specific, we assume that the estimation errors in the traffic matrix are bounded, i.e., if P_i denotes the estimated traffic and $\hat{P}_i$ denotes the actual traffic for OD-flow i, then we have $\forall i, P_i \in [\hat{P}_i(1-\epsilon), \hat{P}_i(1+\epsilon)]$. ϵ quantifies the extent to which the estimated traffic matrix (i.e., our input data) varies with respect to the true traffic matrix. Suppose the optimal sampling strategy for $\hat{P} = \langle \hat{P}_i \rangle_{1 \leq i \leq M}$ is $\hat{d} = \langle \hat{d}_{ij} \rangle_{1 \leq i \leq M, 1 \leq j \leq N}$, and that the optimal sampling strategy for $P = \langle P_i \rangle_{1 \leq i \leq M}$ is $d^* = \langle d^*_{ij} \rangle_{1 \leq i \leq M, 1 \leq j \leq N}$.

Let us consider $\beta(d,P) = \sum_i P_i \times Coverage_i = \sum_i P_i \times (\sum_j d_{ij})$, the total flow coverage for a P-feasible vector $d' = \langle d'_{ij} \rangle_{1 \le i \le M, 1 \le j \le N}$, i.e., satisfying conditions (1)–(4) for P. Our goal to generate a sampling manifest that is robust to bounded error. In other words, we want to generate a new sampling strategy d' from the previously computed optimal solution d^*, and distribute d' to the routers in the network. We want d' to satisfy two properties: (i) d' is feasible for the true but unknown traffic matrix $\hat{P}$, and (ii) $\beta(d', \hat{P})$ is close to the optimal value $\beta(\hat{d}, \hat{P})$.

To start, consider $\hat{d}$ which satisfies the constraints

$$\forall j, \sum_i \hat{d}_{ij} \hat{P}_i \le R_j. \tag{6}$$

Since $\frac{P_i}{1+\epsilon} \le \hat{P}_i$, we also have the inequality,

$$\forall j, \sum_i \hat{d}_{ij} \frac{P_i}{1+\epsilon} \le R_j. \tag{7}$$

Setting $d'' = \frac{\hat{d}}{(1+\epsilon)}$, we note that by (7), d'' is P-feasible. Therefore,

$$\begin{aligned}
\beta(d^*, P) &\ge \beta(d'', P) \\
&= \sum_i P_i \times \left(\sum_j d''_{ij} \right) \\
&= \sum_i P_i \times \left(\sum_j \frac{\hat{d}_{ij}}{1+\epsilon} \right) \\
&\ge \sum_i \hat{P}_i (1-\epsilon) \left(\sum_j \frac{\hat{d}_{ij}}{1+\epsilon} \right) \\
&= \frac{1-\epsilon}{1+\epsilon} \beta(\hat{d}, \hat{P}).
\end{aligned} \tag{8}$$

Next, consider d^* which satisfies the constraints

$$\forall j, \sum_i d^*_{ij} P_i \le R_j. \tag{9}$$

Since $\hat{P}_i(1-\epsilon) \le P_i$, the following inequality holds:

$$\forall j, \sum_i d^*_{ij} (1-\epsilon) \hat{P}_i \le R_j. \tag{10}$$

Setting $d' = d^*(1-\epsilon)$, we see that d' is $\hat{P}$-feasible. Now,

$$\begin{aligned}
\beta(d', \hat{P}) &= \sum_i \hat{P}_i \times \left(\sum_j d'_{ij} \right) \\
&= \sum_i \hat{P}_i \times \left(\sum_j d^*_{ij}(1-\epsilon) \right) \\
&\geq \sum_i \frac{P_i}{1+\epsilon} \times \left(\sum_j d^*_{ij}(1-\epsilon) \right) \\
&= \frac{1-\epsilon}{1+\epsilon} \beta(d^*, P) \\
&\geq \left(\frac{1-\epsilon}{1+\epsilon} \right)^2 \beta(\hat{d}, \hat{P}) \text{ , From Eq 8.}
\end{aligned} \tag{11}$$

If we denote by $\alpha(d, P)$ the minimum fractional coverage objective, we can show by a similar argument that

$$\alpha(d', \hat{P}) = (1-\epsilon)\alpha(d, P) \geq \frac{1-\epsilon}{1+\epsilon} \alpha(\hat{d}, \hat{P}) \tag{12}$$

We note that these bounds are conservative; we will revisit these bounds (particularly Eq. 11) in Section 5.3.

4.6 Robustness to changing traffic matrices

OD-traffic matrices are known to be dynamic as a result of changes in the temporal and spatial aspects of the traffic that traverses a network. These changes in traffic are generally not captured by the bounded error model considered in Section 4.5. We outline our approach for handling such changes.

Long-term variations: Measured backbone network traffic and OD-flows exhibit pronounced but highly predictable time-of-day and day-of-week effects which constitute a major portion of the variations associated with actual OD traffic matrices (e.g., [36]). A common approach for handling these predictable traffic variations is the effective use of historical data. For example, when computing the sampling manifest for, say, this week's Fri. 9am-10am period, we use the OD traffic matrix observed during the previous week's Fri. 9am-10am period as input data.

Short-term variations: To handle less predictable short-term traffic variations, we observe that using traffic matrices averaged over long periods (e.g., week) runs the risk of *under-fitting*; that is, important structure that is present over shorter time scales gets lost due to averaging. On the other hand, traffic matrices that are averaged over short periods (e.g., 5-min intervals) may result in *over-fitting*; that is, accounting for details that are specific to the period in question. As a compromise, we suggest a heuristic approach to handling short-term traffic variations that exploits two distinct time sales. A coarse time scale (e.g., hour) for averaging historical data, and a fine time scale (e.g., 5-min) for running the Coordinated Sampling scheme.

Suppose we are interested in computing sampling manifests for every 5-min interval for the Fri. 9am-10am period of the current week. To avoid over-fitting, we do not use the OD traffic matrices observed during the corresponding 5-min intervals that make up the previous week's Fri. 9am-10am period. Instead, we take the OD traffic matrix obtained by averaging over the previous week's Fri. 9am-10am period, divide it by 12 (the number of 5-min segments per hour), and use the resulting OD traffic matrix P^{old} as input data for computing the sampling manifest for the first 5-min period. At the end of this period, we collect flow data from the individual routers, and using the observed measurements, we obtain the traffic matrix P^{obs}. (For OD-flow i, if the fractional coverage with the current sampling strategy is $Coverage_i$ and x_i sampled flows are reported, then $P_i^{obs} = \frac{x_i}{Coverage_i}$, i.e., normalizing the number of sampled flows by the total flow sampling rate.)

We check if there exist significant differences between the observed traffic matrix P^{obs} and the input data P^{old}. Let $\delta_i = \mathbf{abs}((P_i^{obs} - P_i^{old})/(P_i^{old}))$ denote the estimation error for OD-flow i. If for some OD-flow i, δ_i exceeds a tolerance threshold Δ, then we compute a new traffic matrix entry P_i^{new} for this OD-flow. We use the resulting OD traffic matrix P^{new} as the input for obtaining the sampling manifest for the next 5-min period. We compute P^{new} using the following *conservative update* policy. If P_i^{obs} is greater than P_i^{old} then we set $P_i^{new} = P_i^{obs}$. If P_i^{obs} is smaller than P_i^{old}, then we check the resource utilization of the routers currently responsible for monitoring the OD-flow i. If all these routers have residual resources available, we set $P_i^{new} = P_i^{obs}$; otherwise we set $P_i^{new} = P_i^{old}$.

The rationale behind this conservative update heuristic is that if a router runs out of memory, it may result in underestimating OD-flows for which it is responsible (i.e., P^{obs} is an under-estimate of the actual OD traffic matrix). By updating P^{new} with P^{obs} for such OD-flows, it is likely we would cause a recurrence of the same overflow condition in the next 5-min period. Instead, we err on the side of over-estimating the traffic for each OD-flow. This ensures that the information we obtain for the next period is more reliable and can help us make a better decision when computing the sampling manifest for subsequent 5-min periods. The only caveat of such a heuristic is that we may get a lower effective coverage because we are over-estimating the total traffic volume. Our evaluations with real traffic traces (Section 5.3) show that this performance penalty is low and the heuristic provides near-optimal traffic coverage.

5 Evaluation

In this section we first evaluate the benefits of Coordinated Sampling under ideal conditions (i.e., static network, exact knowledge of input data) and then study its robustness under dynamic traffic conditions. We also describe three representative applications.

5.1 Input data

We implemented a packet-level network simulator to evaluate the performance of different sampling approaches. The simulator takes in as input the sampling algorithm and associated parameters, the network topology and routing matrix (for specifying the set of OD-flows and their routing paths), the OD-level traffic matrix, and the IP flow-size distribution. We use real topologies from

educational backbones [21, 18] and PoP-level topologies inferred from Rocketfuel [42]. For each topology, we construct OD-flows by considering all possible PoP-pairs and determine for each pair the corresponding PoP-level paths. For Internet2 and GÉANT we rely on the publicly available static IS-IS weights and for the Rocketfuel-based topologies we use the inferred link weights [31] to obtain the shortest-path route for each OD-flow.

Topology	PoPs	OD-flows	Flows($\times 10^6$)	Pkts ($\times 10^6$)
AS7018	115	13225	80	320
AS2914	70	4900	51	204
AS3356	63	3969	46	196
AS1239	52	2704	37	148
AS1221	44	1936	32	128
AS3257	41	1681	32	218
GÉANT	22	484	16	64
Internet2	11	121	8	32

Table 1: Parameters for the experiments

Due to lack of publicly available traffic matrix and traffic volume information for the commercial ISPs, we take the following approach. Taking 8 million IP flows (per 5-minute interval)[5] as the baseline traffic volume for Internet2, for each the other topologies, we scale the total traffic (number of IP flows) by the number of PoPs in the topology. We believe that these traffic volumes are of the same order of magnitude as the estimates reported for Tier-1 backbones. Table 1 summarizes the various topologies. To obtain the traffic matrices, we first annotate each PoP in the topology with the population p_i of the city it is located in. Then we use a simple gravity-model [39] to obtain the traffic volume for each OD-flow; that is, we assume that the total traffic between PoPs i and j is proportional to $p_i \times p_j$. We assume that flow size measured in number of packets is Pareto-distributed, i.e., $Prob(Flowsize > x\ packets) = (\frac{c}{x})^{\alpha}, x \geq c$ with $\alpha = 1.8$ and $c = 4$.[6]

5.2 Benefits of Coordinated Sampling

We compare the benefits of Coordinated Sampling against (i) uniform packet sampling, (ii) uniform packet sampling at ingress and egress nodes only, (iii) random flow sampling, and (iv) optimal uncoordinated flow sampling. Table 2 presents a taxonomy of the spectrum of sampling alternatives we consider[7].

Coordinated Sampling and flow sampling are constrained by the amount of SRAM on each router[8]. We assume that each PoP in the network is provisioned hold up to 200,000 flow records.

[5]The weekly aggregate traffic on Internet2 is roughly 175TB. Ignoring time-of-day and effects, this translates into 0.08TB per 5-minute interval. Assuming an average flow size of 10KB, this translates into roughly 8 million flows.

[6]We use these as representative values. Our results are similar across a range of flow size parameters.

[7]We do not consider optimal network-wide uniform packet sampling [7]. Instead, we consider optimal uncoordinated flow sampling as a hypothetical flow-sampling extension to Cantieni et al. [7].

[8]Since Sharma and Byers [39] assume flow-sampling without imposing any SRAM constraints, it is not possible to present a direct quantitative comparison between Coordinated Sampling and their approach.

Sampling Method	Flow vs. Packet	Coordi--nated	Resource Limits	Network Wide
Uniform packet sampling	Packet	No	No	No
Edge Uniform pkt sampling	Packet	No	No	No
Flow sampling	Flow	No	Yes	No
Optimal uncoordinated flow sampling	Flow	No	Yes	Yes
Coord. Sampling	Flow	Yes	Yes	Yes

Table 2: Taxonomy of sampling alternatives

Even assuming a conservative estimate of 32 bytes for each flow entry [14], this translates into a requirement of only $200,000 \times 32 = 6.4$MB of SRAM per PoP. For uniform packet sampling, we assume a sampling rate of 0.01 and impose no memory constraints on the routers [14]. For the edge-based uniform packet sampling case that may reflect a feasible and practical alternative for some ISPs [15], we assume a sampling rate of 0.02 and impose no memory constraints on the routers. For random flow sampling, we assume that every node uses a uniform flow-sampling rate of 0.01. In the case of optimal uncoordinated flow sampling, the flow sampling rates are chosen such that each node maximally utilizes its available memory.[9]

Coverage Benefits: Figure 3 compares the total flow-coverage obtained with the different sampling schemes for the PoP-level topologies in Table 1. We observe that random flow sampling results in less flow coverage than the uniform packet sampling alternatives (i) and (ii). This is a direct consequence of the resource constraints associated with flow sampling. Also note that using a higher sampling rate of 0.02 for edge-based uniform packet sampling only marginally improves flow coverage over (i). Relying on the network-wide but uncoordinated sampling approach (iv) for setting flow sampling rates can provide substantial improvements (up to 75%) over (i) and (iii). However, we can boost these improvements even further (up to 100%) by using Coordinated Sampling.

Figure 4 compares the minimum fractional coverage per OD-flow obtained by different sampling strategies. We see that Coordinated Sampling outperforms all alternatives by a substantial margin, including the optimized uncoordinated flow-sampling scheme (iv). This ability to specify and attain network-wide monitoring objectives is a key strength of our approach. Two other observations are worth noting. First, the minimum fractional coverage is much less (more than $2\times$ in some cases) than the total coverage. Second, the differences between the various topologies in terms of the minimum fractional coverage are more pronounced than in terms of total coverage.[10] The reason for these observations is the structure of the traffic matrix. Specifically, we observe

[9]The flow sampling rate for a node is $min(1, \frac{M}{T})$, where $M = 200000$ and T is the total number of flows the node observes.

[10]Note that AS7018, Internet2, and GÉANT are distinctively better with respect to minimum fractional coverage than AS1221 and AS3356, even though the traffic volumes scale linearly with the number of PoPs

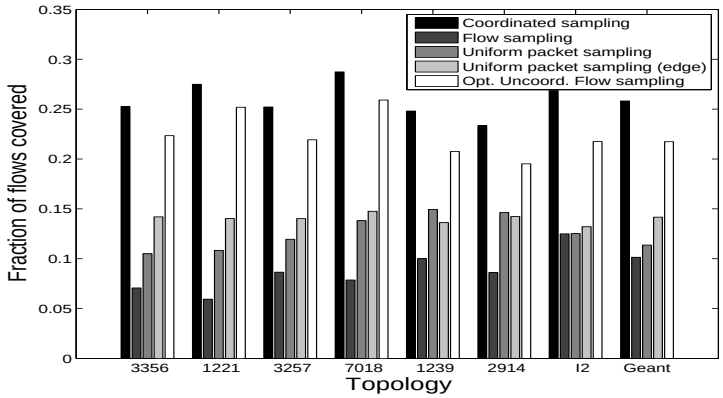


Figure 3: Total flow coverage

that the presence of disproportionately large diagonal and off-diagonal elements in a traffic matrix becomes a dominant factor in determining the minimum fractional coverage that is feasible given the resource constraints.[11]

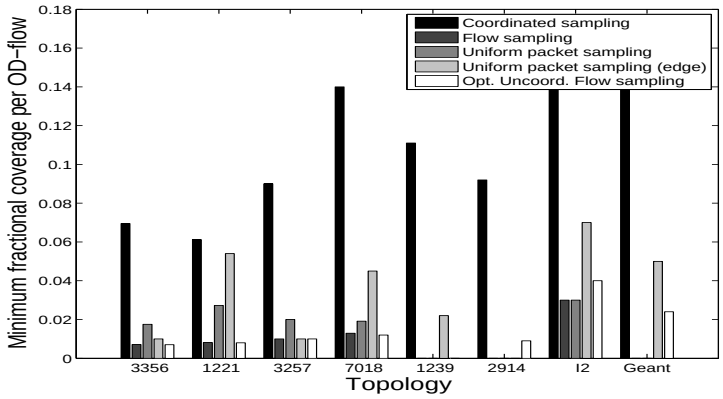


Figure 4: Min. fractional coverage per OD-flow

Reporting Benefits: In Figure 5, we show the *wasted bandwidth* as a fraction of the number of duplicate flow reports (due to multiple routers monitoring the same flows) to the number of useful (i.e., distinct) flow reports. The absence of an entry for Coordinated Sampling in Figure 5 reflects our design: by assigning non-overlapping hash-ranges to individual monitors, we avoid duplicate sampling of traffic flows. In addition to wasting reporting bandwidth, these duplicate reports can also induce operational difficulties in managing and mining the data collected from multiple monitors. We observe that network-wide uncoordinated flow sampling results in the largest amount of duplicate flow-reports (as high as 30%), while uniform packet sampling can result in up to 14% duplicate reports. Using edge-based uniform packet sampling can alleviate this waste to some extent, since redundant reporting from non-terminal (i.e., transit) routers is avoided.

[11] As an example, AS1221 (Telstra) has PoPs in major Australian cities and in Los Angeles. The bias in the population distribution across PoPs is such that the top-4 PoPs (Sydney, Melbourne, Los Angles, Sydney) account for more than 60% of the total traffic volume and routes between these cities do not go through any other PoPs.

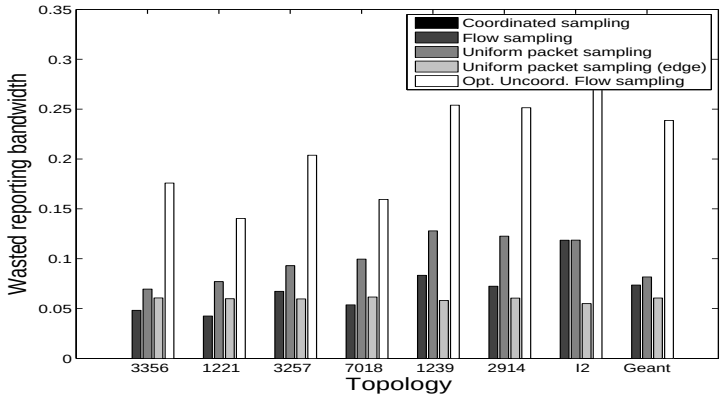


Figure 5: Duplicate reporting bandwidth

The *maximum reporting bandwidth* on any single PoP is shown in Figure 6. We normalize the reporting bandwidth by the bandwidth required for Coordinated Sampling. The reporting bandwidth for Coordinated Sampling and flow sampling is bounded by the amount of memory that the routers are provisioned with – memory relates directly to the number of flow-records that a router needs to export. Figure 6 shows that the maximum reporting bandwidth for uniform packet sampling can be as high as 7-10 times the reporting bandwidth required for Coordinated Sampling. This suggests that our approach has the added benefit of avoiding reporting hotspots by efficiently assigning monitoring responsibilities across routers in such way that each operates within the specified resource limits.

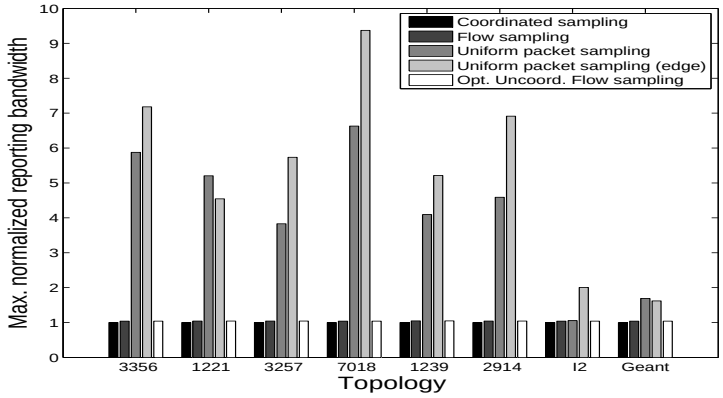
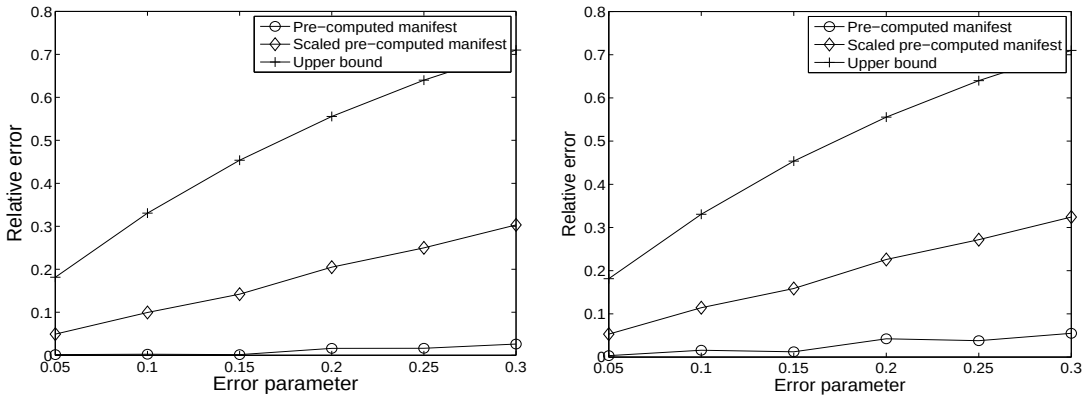


Figure 6: Max. reporting bandwidth per-PoP

5.3 Robustness Properties

Inaccurate traffic matrices: To study the robustness of Coordinated Sampling to inaccuracies in traffic matrix estimates, we consider the Internet2 topology and use a gravity model as an approximation of its exact but unknown baseline OD-level traffic matrix (e.g., see [35]). We use the error model discussed in Section 4.5: if $\hat{P}_i$ denotes the exact but unknown traffic volume (in number

of IP-flows) for OD-flow i, then the estimated traffic volume P_i used as input to our approach is drawn uniformly at random from the interval $[\hat{P}_i(1-\epsilon), \hat{P}_i(1+\epsilon)]$.



(a) Mean error

(b) Maximum error

Figure 7: Sensitivity of the total coverage as a function of the error in the input traffic matrix used for computing a sampling strategy

We are interested in the relative error between the optimal sampling strategy (computed using the true but unknown traffic matrix $\hat{P}$) and the sampling manifest derived using the inaccurate estimated traffic matrix P. Figure 7 shows the mean and maximum relative error (over 20 independent runs), as function of ϵ, for total flow coverage. The figure shows three relative error curves: (i) theoretical upper bound from Section 4.5[12], (ii) performance of the sampling manifest based on the inaccurate input data, and (iii) performance of the sampling strategy obtained by scaling the sampling manifest in (ii) by a factor $1-\epsilon$ (Section 4.5). We observe that the total flow coverage provided by Coordinated Sampling is remarkably insensitive to inaccuracies in the input data; even with errors as high as 30%, the relative error with respect to the optimal solution is less than 5%.[13] The figure suggests that the lower bounds are conservative, and that we can expect much better performance in practice. Also, since estimates of the large traffic matrix elements have been shown to be significantly more accurate than estimates of the small elements [45], we expect the robustness of Coordinated Sampling to errors associated with estimated OD-traffic matrices to be even better in practice.

Changing Traffic matrices: To explore the robustness to realistic changes of traffic matrices, we consider a two-week snapshot (Dec 1–14, 2006) of flow data from Internet2. The flow data is collected using uniform packet sampling with a sampling rate of 1-in-100 packets. We map each flow entry to the corresponding network ingress and egress points using the technique outlined in

[12]Eq. 11 proves $\beta(d', \hat{P}) \geq (\frac{1-\epsilon}{1+\epsilon})^2 \beta(\hat{d}, \hat{P})$. The theoretical upper bound on the relative error will be $\frac{\beta(\hat{d},\hat{P})-\beta(d',\hat{P})}{\beta(\hat{d},\hat{P})} \leq \frac{4\epsilon}{(1+\epsilon)^2}$.

[13]Similar results hold for the minimum fractional coverage metric, except that the worst case error can be quite large (between 20-30% for errors in the 20-30% range).

Feldmann et al. [15].[14] We assume that there are no routing changes in the network, and that the sampled flow records represent the actual traffic in the network (since Coordinated Sampling does not suffer from flow size biases there is no need to renormalize the flow sizes by the sampling rate). Since the sampled data contains only two million distinct flows on average, we scale down the per-PoP memory by a factor of 4 from 200,000 (from Section 5.2) to 50,000 flow records.

To compute the sampling manifest for a particular period for the current week, we use the previous week's flow data measured for that same period to obtain the estimated OD traffic matrix. Figure 8 compares the total flow coverage obtained with different strategies for using the historical data to the optimal solution (i.e., assuming perfect traffic information in the sense that the traffic matrix is computed with the actual flow data for the current interval). As expected, the optimal flow coverage exhibits the same time-of-day and day-of-week effects as the traffic matrices themselves. For example, during the weekend (day2 and day3), we can get up to 70% coverage compared to the weekdays, when the coverage is typically in the 20-50% range. We also notice that using coarse-grained historical information (i.e., daily or weekly averages) gives sub-optimal solutions. On the other hand, relying on traffic matrices that are based on hourly averages from the previous week gives near-optimal total flow coverage and seems to represent a time scale of practical interest that avoids both the risk of over-fitting as well as the risk of under-fitting. In contrast, for the minimum fractional coverage per OD-flow, Figure 9 shows that using the per-hour estimates, we get less than half the optimal minimum fractional coverage for many of the 5-minute time-slots. This is primarily because of short-term variations that the historical traffic matrices cannot account for.

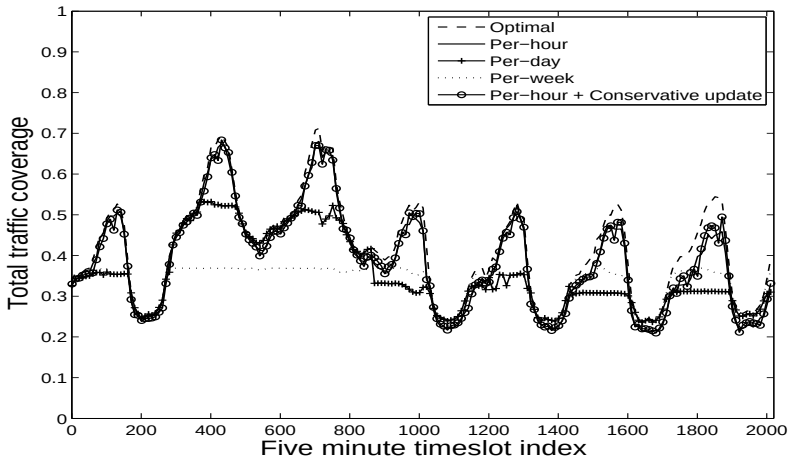


Figure 8: Comparing total traffic coverage with different approaches for selecting historical traffic matrices for computing a Coordinated Sampling strategy.

Figures 8 and 9 also depict a curve labeled "Per-hour + Conservative update" that results from using the heuristic for dealing with short-term traffic variations described in Section 4.6 (we use $\Delta = 0.1$). We observe that the heuristic can significantly improve the performance in the case of the minimum fractional coverage metric, and achieves near-optimal performance for the total traffic coverage as well. While our approach needs further analysis, these results demonstrate the

[14]Since IP-addresses are anonymized by zero-ing out the last 11 bits, there is some ambiguity associated with the egress resolution, but this does not introduce a significant bias as less than 3% of the flows are affected.

promise of using historical per-hour traffic matrices combined with a conservative update heuristic for handling both the expected long-term and unexpected short-term traffic variations.

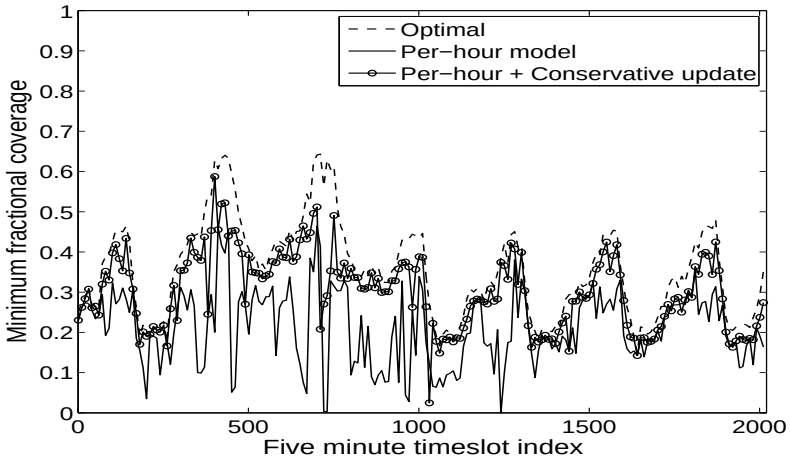


Figure 9: Comparing the minimum fractional coverage to the optimal solution with the conservative update function.

5.4 Applications

To illustrate that Coordinated Sampling supports a wide range of tasks of interest to ISPs, we consider three representative applications: traffic engineering (volume estimation), security (scanner detection), and network provisioning.

Traffic Volume Estimation: Many traffic engineering and accounting applications are interested in the packet and byte volumes per OD-flow. Here, we focus on obtaining packet-count estimates for each OD-flow. (We do not compare the byte counts since uniform packet sampling has additional packet-size biases that flow sampling does not suffer from [14].) To this end, we need accurate packet-level data, and since Internet2 flow data has biases due to packet sampling, we use our simulation results for the Internet2 topology (Section 5.2). For both uniform packet sampling and edge-based uniform packet sampling, the estimates are obtained using the method suggested by Duffield et al. [12]. For Coordinated Sampling, we identify the fractional flow coverage $Coverage_i = \sum_j d_{ij}$ for OD-flow i and renormalize the total packet volume by this factor (this is an unbiased estimate of the total traffic volume). Figure 10 shows the CDF of the relative error (we consider only the magnitude of the relative error, not whether it is positive or negative) in estimating the traffic volume on each OD-flow. We observe that Coordinated Sampling results in traffic volume estimates that are comparable to or even better than those obtained using uniform packet sampling. This illustrates that Coordinated Sampling does not impair the accuracy required by traditional traffic engineering applications.

Scanner Detection: We take a five-minute trace from the Internet2 dataset and treat each flow record as a single packet. (Note that by ignoring flow sizes we can only overestimate the performance of packet sampling.) Using this to serve as the background traffic, we inject traffic records simulating the presence of 1000 scanners (distributed at random in the network) and consider a

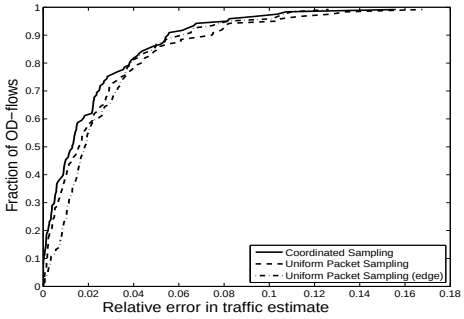


Figure 10: Relative error in volume estimation

threshold-based scan detection approach: we flag any host that contacts more than k distinct destination IP addresses in the sampled data. Figure 11 shows the ROC-curve for both uniform packet sampling (with 1-in-50 sampling) and Coordinated Sampling for two scenarios. In the first scenario, each scanner generates 100 scans (scan destinations are selected uniformly at random within the trace) in the five-minute interval and in the second scenario each scanner generates 200 scans. Each point on the ROC-curve represents the false positive and false negative rate for a fixed detection threshold k. We vary k between 1 and 80 in this experiment. For lower values of k we expect the false negative rate (i.e., not detecting a scanner) to be low but the false positive rate (i.e., flagging a host which is not one of the scanners) to be high. As k increases, the false positive decreases, but there is an increase in the false negative rate. Ideally, we want the ROC-curve to have a low false positive rate and a low false negative rate. We observe that the ROC-curves for Coordinated Sampling show significantly better performance than those for uniform packet sampling (i.e., the curves for Coordinated Sampling are closer to the origin).

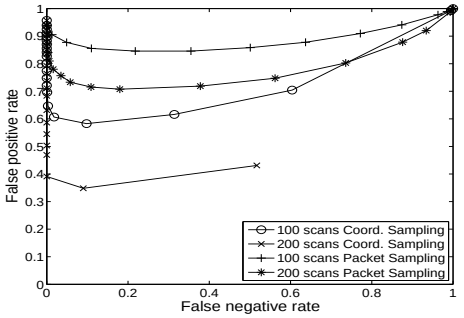


Figure 11: ROC-curve for scanner detection

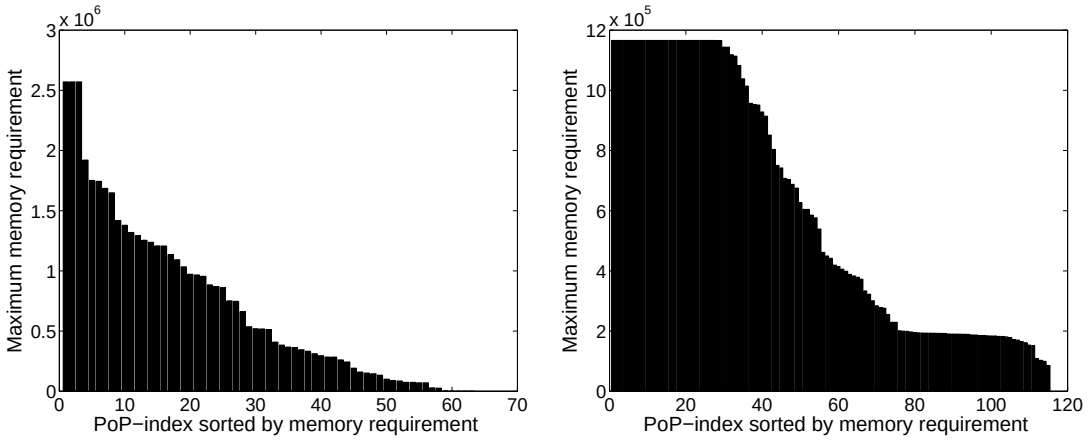
Network provisioning: An alternative version of the network-wide formulation (Section 3.3.2) can be posed as a capacity provisioning problem; i.e., how should a network operator invest resources at routers (e.g., memory) to achieve a given target traffic coverage? To discuss such a "what-if" scenario, we use the notation and formulation from Section 3.3.2 and let α_i denote the targeted fraction of traffic on OD-flow i to be monitored; that is,

$$\forall i,\ Coverage_i \geq \alpha_i$$

The monitoring load L_j on router j is given by

$$\forall j,\ L_j = \sum_i d_{ij} \times P_i$$

and translates directly into the memory and reporting bandwidth that need to be provisioned on the router. It also reflects the cost incurred by the operators (e.g., memory upgrades on router hardware). We consider the following objective: minimizing the maximum load on any single router in the network.



(a) AS 3356 (b) AS 7018

Figure 12: Distribution of memory requirement across PoPs

Across the different PoP-level topologies we find that even with a target flow coverage of 90%, the maximum memory required per PoP is of the order of a 1-3 million traffic records. Assuming a 32-byte flow record, this translates into a maximum memory requirement of 90MB per-PoP, which is larger than the memory capacities on routers today, but not technologically inconceivable. This is promising in view of certain applications for which near-complete traffic coverage is desirable (e.g., forensic applications [43]). Figure 12 shows the distribution of the per-PoP memory requirement (in terms of number of flow records). We observe that the number of nodes that need very high provisioning is small. This is consistent with the observations in Section 5.2 regarding the structure of the underlying traffic matrix – dominant PoPs that carry a significant fraction of the traffic naturally demand better provisioning than smaller PoPs.

6 Summary and Future Work

Compared to current solutions, Coordinated Sampling offers several advantages. First, by increasing flow coverage more than two-fold, it provides high fidelity for new kinds of security applications, without compromising the accuracy required by more traditional traffic engineering

applications. Second, it allows operators to specify and achieve fine-grained network-wide monitoring objectives. For example, it allows operators to achieve an order of magnitude improvement in the minimum fractional coverage per OD-flow thereby providing network-wide visibility. Third, through hash-based coordination, it allows operators to efficiently leverage available monitoring capacity in the network without requiring expensive distributed protocols and communication between routers.

Routers only need to implement a very simple algorithm (Figure 2) and do not have to engage in distributed computations or communicate with each other to obtain their logging responsibilities. The complexity is in obtaining the configuration files that spell out in detail the sampling instructions for the different routers in the network. However, this decision logic will necessarily be implemented in a centralized processing facility, similar in spirit to recent proposals that argue in favor of more centralized network management [6, 19, 3]. Coordinated Sampling thus also matches well with the current trends toward a more centralized operation model of ISPs.

Our analysis and evaluations demonstrate that Coordinated Sampling possesses attractive robustness properties with respect to realistic network conditions (e.g., inaccuracies in the input data, temporal and structural changes of network traffic). An aspect of robustness that has not been addressed in this paper concerns the number of reconfigurations under traffic dynamics. To reduce management complexity, network operators may prefer sampling manifests that are stable over time or require only a handful of reconfigurations in response to some of the typical events they expect. Here, a reconfiguration refers to either (i) a non-zero d_{ij} value becoming zero in the new sampling strategy recomputed after the traffic change, or (ii) a d_{ij} entry that was previously zero becoming non-zero in the new sampling strategy. As a preliminary exploration, we augmented the objective function with a reconfiguration cost term. The reconfiguration cost penalizes feasible sampling strategies that, while optimal otherwise, require a large number of reconfigurations when compared to the sampling strategy currently in use. Figure 13 shows the results of this preliminary exploration using data from Internet2 (we only show the results for day2 from week2; results for other days were similar). We see that the new sampling manifests are relatively stable throughout the 24-hour period and require in general only a small number of reconfigurations (on average less than 5% of entries). Moreover, this added robustness feature is achieved with negligible loss in total flow coverage and minimum fractional coverage (0.5% and 3% respectively) (not shown). These preliminary results are similar to prior work on configuring link weights in the context of intra-domain routing [1, 17]. A promising avenue of future work is exploring this connection and developing strategies that are explicitly designed to have as few reconfigurations as possible.

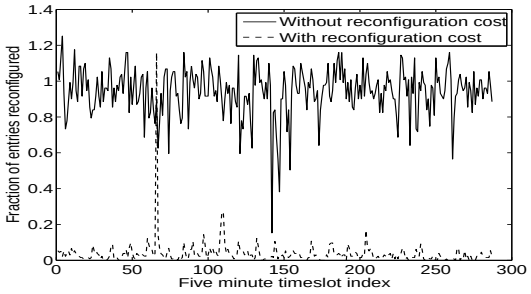


Figure 13: Effect of introducing reconfiguration cost to the formulation

The other dimension of robustness not addressed in this paper is with respect to routing dynamics caused by node and link failures. Since our approach is ISP-centric, we propose the following approach. One common network management task is to ensure smooth network operations in the case of critical router or link failures. This is achieved by using the network configuration and estimated traffic matrix, simulating particular failure events, and precomputing new set of link weights in such a way that under the particular failure scenario, the remaining network can handle the (rerouted) traffic without problems [16, 17]. As a by-product of this traffic engineering exercise, we can precompute the optimal sampling scheme for the scenario corresponding to each particular failure event (i.e., using the appropriate mapping of OD-flows to routers and the traffic matrix that reflects the rerouting of traffic) and have it ready when this failure actually occurs.

A natural extension for exploring the virtues of Coordinated Sampling would be using router-level ISP topologies, where the role (e.g., backbone, edge, access) and specifications of each individual router are known. However, actual ISP router-level topologies are generally not available and inferred topologies (e.g., [42]) lack the annotations necessary for our purposes (e.g., identifying gateway and backbone routers). We expect the benefits of Coordinated Sampling compared to alternative sampling strategies to be even better on router-level topologies for two reasons. First, since router-level topologies are more fine-grained than PoP-level topologies we expect greater benefits from coordination (e.g., more routers per-path). Second, our approach has the ability to efficiently exploit the increased heterogeneity provided by router-level topologies as far as individual router capabilities, OD route diversity, and OD traffic demand patterns are concerned. One direction of future work is to build on the work of Li et al. [29] and study the properties of Coordinated Sampling as a function of the granularity of the underlying topology.

References

[1] D. Applegate and E. Cohen. Making intra-domain routing robust to changing and uncertain traffic demands: Understanding fundamental tradeoffs. In *Proc. of ACM SIGCOMM*, 2003.

[2] E. B. Claise. Cisco Systems NetFlow Services Export Version 9. RFC 3954, 1990.

[3] H. Ballani and P. Francis. CONMan: Taking the compexity out of network management . In *ACM SIGCOMM INM*, 2006.

[4] S. Bellovin, M. Leech, and T. Taylor. ICMP traceback messages. Internet draft, work in progress, 2001.

[5] D. Brauckhoff, B. Tellenbach, A. Wagner, A. Lakhina, and M. May. Impact of traffic sampling on anomaly detection metrics. In *Proc. of IMC*, 2006.

[6] M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, and J. van der Merwe. Design and implementation of a routing control platform. In *Proc. of NSDI*, 2005.

[7] G. R. Cantieni, G. Iannaccone, C. Barakat, C. Diot, and P. Thiran. Reformulating the monitor placement problem: Optimal network-wide sampling. In *Proc. of CoNeXT*, 2006.

[8] M. Collins and M. K. Reiter. Finding peer-to-peer file-sharing using coarse network behaviors. In *Proc. of ESORICS*, 2006.

[9] N. Duffield and M. Grossglauser. Trajectory sampling for direct traffic observation. In *Proc. of ACM SIGCOMM*, 2001.

[10] N. Duffield, C. Lund, and M. Thorup. Charging from sampled network usage. In *Proc. of IMW*, 2001.

[11] N. Duffield, C. Lund, and M. Thorup. Learn more, sample less: Control of volume and variance in network measurement. *IEEE Transactions in Information Theory*, 51(5):1756–1775, 2005.

[12] N. Duffield, C. Lund, and M. Thorup. Optimal combination of sampled network measurements. In *Proc. of IMC*, 2005.

[13] C. Estan, K. Keys, D. Moore, and G. Varghese. Building a better netflow. In *Proc. of ACM SIGCOMM*, 2004.

[14] C. Estan and G. Varghese. New directions in traffic measurement and accounting. In *Proc. of ACM SIGCOMM*, 2002.

[15] A. Feldmann, A. G. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True. Deriving traffic demands for operational IP networks: methodology and experience. In *Proc. of ACM SIGCOMM*, 2000.

[16] B. Fortz, J. Rexford, and M. Thorup. Traffic Engineering with Traditional IP Routing Protocols. IEEE Communications Magazine, Oct. 2002.

[17] B. Fortz and M. Thorup. Optimizing OSPF/IS-IS Weights in a Changing World. *IEEE Journal on Selected Areas in Communications*, 20(4), May 2002.

[18] GÉANT. `http://www.geant.net`.

[19] A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Meyers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang. A Clean Slate 4D Approach to Network Control and Management. *ACM SIGCOMM CCR*, 35(5), Oct. 2005.

[20] N. Hohn and D. Veitch. Inverting sampled traffic. In *Proc. of IMC*, 2003.

[21] Internet2. `http://www.internet2.edu`.

[22] R. Kompella and C. Estan. The power of slicing in internet flow measurement. In *Proc. of IMC*, 2005.

[23] A. Kumar, M. Sung, J. Xu, and J. Wang. Data streaming algorithms for efficient and accurate estimation of flow distribution. In *Proc. of ACM SIGMETRICS*, 2004.

[24] A. Lakhina, M. Crovella, and C. Diot. Diagnosing network-wide traffic anomalies. In *Proc. of ACM SIGCOMM*, 2004.

[25] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. Kolaczyk, and N. Taft. Structural analysis of network traffic flows. In *Proc. of ACM SIGMETRICS*, 2004.

[26] A. Lall, V. Sekar, J. Xu, M. Ogihara, and H. Zhang. Data streaming algorithms for estimating entropy of network traffic. In *Proc. of ACM SIGMETRICS*, 2006.

[27] S. Lee, T. Wong, and H. Kim. Secure split assignment trajectory sampling: A malicious router detection system. In *Proc. of IEEE/IFIP DSN*, 2006.

[28] J. Li, M. Sung, J. Xu, L. Li, and Q. Zhao. Large-scale IP Traceback in High-speed Internet: Practical Techniques and Theoretical Foundation. In *Proc. of IEEE Symposium of Security and Privacy*, 2004.

[29] L. Li, D. Alderson, W. Willinger, and J. Doyle. A first-principles approach to understanding the internet's router-level topology. In *Proc. of ACM SIGCOMM*, 2004.

[30] X. Li, F. Bian, H. Zhang, C. Diot, R. Govindan, W. Hong, and G. Iannaccone. MIND: A Distributed Multidimensional Indexing for Network Diagnosis. In *Proc. of the IEEE INFOCOM*, 2006.

[31] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson. Inferring link weights using end-to-end measurements. In *Proc. of IMW*, 2002.

[32] J. Mai, C.-N. Chuah, A. Sridharan, T. Ye, and H. Zang. Is sampled data sufficient for anomaly detection. In *Proc. of IMC*, 2006.

[33] A. Mankin, D. Massey, C.-L. Wu, S. F. Wu, and L. Zhang. On design and evaluation of "intention-driven" icmp traceback. In *Proc. of IEEE ICCCN*, 2001.

[34] M. Ramakrishna, E. Fu, and E. Bahcekapili. Efficient hardware hashing functions for high performance computers. *IEEE Transactions on Computers*, 46(12):1378–1381, 1997.

[35] M. Roughan. Simplifying the synthesis of internet traffic matrices. *ACM SIGCOMM CCR*, 35(5), 2005.

[36] M. Roughan, A. Greenberg, C. Kalmanek, M. Rumsewicz, J. Yates, and Y. Zhang. Experience in Measuring Internet Backbone Traffic Variability:Models, Metrics, Measurements and Meaning. In *Proceedings of International Teletraffic Congress (ITC)*, 2003.

[37] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical Network Support for IP Traceback. In *Proc. of ACM SIGCOMM*, 2000.

[38] V. Sekar, N. Duffield, K. van der Merwe, O. Spatscheck, and H. Zhang. LADS: Large-scale Automated DDoS Detection System. In *Proc. of USENIX ATC*, 2006.

[39] M. R. Sharma and J. W. Byers. Scalable coordination techniques for distributed network monitoring. In *Proc. of PAM*, 2005.

[40] S. Singh, C. Estan, G. Varghese, and S. Savage. Automated worm fingerprinting. In *Proc. of OSDI*, 2004.

[41] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, and W. T. Strayer. Hash-Based IP Traceback . In *Proc. of ACM SIGCOMM*, 2001.

[42] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP Topologies with Rocketfuel. In *Proc. of ACM SIGCOMM*, 2002.

[43] Y. Xie, V. Sekar, D. A. Maltz, M. K. Reiter, and H. Zhang. Worm origin identification using random moonwalks. In *Proc. of IEEE Symposium on Security and Privacy*, 2005.

[44] X. Yang, D. Wetherall, and T. Anderson. A DoS-limiting Network Architecture. In *Proc. of ACM SIGCOMM*, 2005.

[45] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg. Fast accurate computation of large-scale IP traffic matrices from link loads. In *Proc. of ACM SIGMETRICS*, 2003.

[46] Q. Zhao, Z. Ge, J. Wang, and J. Xu. Robust traffic matrix estimation with imperfect information: Making use of multiple data sources. In *Proc. of ACM SIGMETRICS*, 2006.

[47] T. Zseby, M. Molina, N. Duffield, and S. Niccolini. Sampling and Filtering Techniques for IP Packet Selection. IETF Draft http://www.ietf.org/internet-drafts/draft-ietf-psamp-sample-tech-07.txt, 2005.